



# A heuristic for the retrieval of objects in low resolution video

Fanny Chevalier, Maylis Delest, Jean-Philippe Domenger

## ► To cite this version:

Fanny Chevalier, Maylis Delest, Jean-Philippe Domenger. A heuristic for the retrieval of objects in low resolution video. Workshop on Content-Based Multimedia Indexing, Jun 2007, Talence, France. pp.144-151. hal-00306683

**HAL Id: hal-00306683**

**<https://hal.science/hal-00306683>**

Submitted on 28 Jul 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A HEURISTIC FOR THE RETRIEVAL OF OBJECTS IN LOW RESOLUTION VIDEO

*Fanny Chevalier, Maylis Delest and Jean-Philippe Domenger*

LaBRI (University of Bordeaux 1,CNRS)  
{chevalie, maylis, domenger}@labri.fr

## ABSTRACT

In this paper, we tackle the problem of matching of objects in video in the context of the rough indexing paradigm. In this context, the video data are of very low resolution and segmentation is consequently inaccurate. The region features (texture, color, shape) are not strongly relevant due to the resolution. The structure of the objects must be considered in order to improve the robustness of the matching of regions. Indeed, the problem of object matching can be expressed in terms of directed acyclic graph (DAG) matching. Here, we propose a method based on a heuristic in order to approach object matching. The results are compared with those of a method based on relaxation matching.

**Keywords:** video object retrieval, rough indexing paradigm, error-tolerant graph matching, heuristic.

## 1. INTRODUCTION

This paper addresses the problem of object retrieval in video, and more precisely, matching of a moving object extracted from prototype video frame with objects extracted from other frames in a video stream. Typical applications of our method are the retrieval of objects in video-shot collections or grouping of the shots that contain the same protagonist into video scenes. In video, the shape, the size and the structure of objects change mainly due to camera motion, object motion and occlusion phenomena. Thus, the structure of the same object at different times in a video may present significant differences.

Furthermore, our work is placed in the context of the rough indexing paradigm [5, 16, 17]. The data considered in this approach come from partially decoded MPEG compressed streams. Here, we only consider the first coefficient of the discrete cosinus (DC) transform of video frames. The DC-images are composed of color pixels which represent the mean values of  $8 \times 8$  squared blocks in the original video frames. In this way, the colorimetric and geometrical information are strongly smoothed.

An image partition is classically represented by a region adjacency graph (RAG). The RAG modelling allows to express the matching of segmented objects in terms of graph

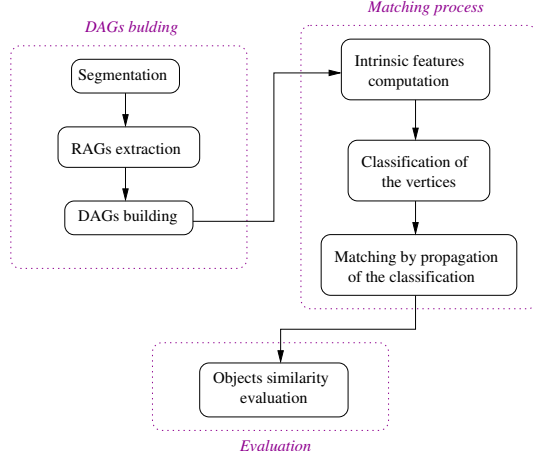
matching. In our context, the segmentation of the same object may strongly differ with time in video due to its motion, occlusions and down sampling discretization. The corresponding RAGs may be strongly different as well. Consequently, an exact graph matching is not efficient [6].

Several techniques for error-tolerant graph matching are frequently used in Content-Based Image Retrieval (CBIR) and are more adequate for video context. Some of them [19, 21] only consider intrinsic metrics (adjacency relations between vertices). Other methods consider a similarity measure between the regions of objects based on region characteristics [14, 20]. These last methods use sophisticated visual descriptors (color, texture, geometry) on regions, as for instance MPEG7 descriptors or color histogram of regions. In our context of rough data, these are not relevant. Therefore, these methods produce matching errors because of the loss of the global object's topology information.

Another kind of graph matching methods uses relaxation techniques [11, 15]. Based on a similarity measure computed between pairs of regions, processes of relaxation implicitly evaluate neighbourhood likeliness to adjust the similarity measure between pairs of regions. In this way, the regions of an object are recognizable even if small local motions of the object or segmentation errors have deformed them. In the rough indexing paradigm, we have proposed a relaxation matching method [5]. The results of this method will be compared with those provided by the method presented here.

In the problem of object matching in video, natural objects are often articulated and even if region characteristics vary with time, the structure of a region neighborhood would remain stable. In this paper, we propose a matching method that takes into account the topology of objects. The matching is based on object structure parts that are quasi-similar in the sense of their RAGs. We also consider the mean color of the regions and their relative area in order to drive the matching process.

An overview of the method is presented in figure 1. The first step consists in building a directed acyclic graph (DAG) associated to each segmented object. Starting from a partition of an object into 4-connected regions, we compute the induced region adjacency graph (RAG). The vertices of the RAG represent the regions belonging to the object and the edges encode the neighborhood relations. Then, we transform



**Fig. 1.** The overall scheme of the method

this RAG into a DAG by orienting its edges. Considering that a region having an important area is more significant, the root of the DAG is associated with the region having the highest area. The edges are oriented from a region to its neighbor regions with a smaller relative area. After this step, each segmented object is associated with one DAG. The second step (see matching process in figure 1) is devoted to the search of a maximal quasi-similar sub-DAG between the DAGs. Intrinsic metrics are computed for each vertex of the DAG which allow us to define a distance between the vertices. After what, we label the vertices of the DAG, such that two vertices with a distance less than a given threshold have the same label. The last step consists in propagating the labels of vertices according to their color similarity and the labelling of children vertices. Then, the vertices belonging to similar sub-DAGs have the same label. At the end, in order to decide if the objects match to each other, we use a similarity measure of objects based on the relative area of the sets of regions associated to the nodes of the similar sub-DAGs.

The paper is organized as follows. In section 2, we briefly introduce segmentation of objects in the rough indexing paradigm and describe how DAGs are built from RAGs. In section 3, we introduce the intrinsic metrics associated with DAG vertices. Section 4 describes the finding of similar sub-DAGs. The object matching algorithm is described in section 5. Results on natural video are presented in section 6 and a conclusion is given in section 7.

## 2. SEGMENTATION AND RAG-BUILDING OF OBJECTS FROM “ROUGH” VIDEO

In this paper, the objects that we consider are obtained as follows: first, a zone of interest that corresponds to foreground objects is extracted from the DC-frame by the computation of the binary motion mask [16]. The pixels that have a local



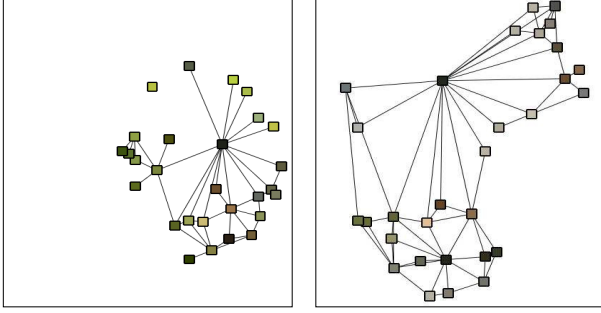
**Fig. 2.** Original video frames with corresponding segmented objects

motion different from the global camera motion belong to the motion mask. Note that the zone of interest is not necessary a connected component. Then, we partition this zone of interest by applying a segmentation process developed in [16].

The pixels of DC-images considered here are the mean color of  $8 \times 8$  squared blocks in original video frames. In DC-images the details of initial images are smoothed by this down-sampling. The segmentation process used in this work is based on a region growing algorithm performed with a modified watershed [16] and is applied only on the region of interest (binary motion mask).

The segmentation process produces a partition  $\mathcal{P}$  of the zone of interest into a set  $\{r_1, \dots, r_n\}$  of 4-adjacent regions that represents a segmented object. Each region is homogeneous according to a colorimetric homogeneity criterion which expresses the difference of color vectors of pixels in a region and the mean color vector of a region compared to a region adaptive threshold [16]. In figure 2, two video frames at different times are shown. The same object (an old man) appears in both frames and the results of the foreground object extraction (binary mask) and its segmentation (partition into regions) are displayed under the original corresponding frames. One can see that many differences exist due to scale deformation, local motions (e.g. the man’s arm), partial occlusion and additional background pixels.

In a classical way, we associate a RAG  $G(G_V, G_E)$  (where  $G_V$  is the set of vertices and  $G_E$  are the edges of the RAG), to a partition  $\mathcal{P} = \{r_1, \dots, r_n\}$ . Each region  $r_i \in \mathcal{P}$  is considered as a vertex  $s_i$  of  $G_V$ . We denote by  $R(s)$  the region  $r$  that is represented by the vertex  $s$  in the RAG. By extension, if  $S$  is a set of vertices,  $R(S)$  corresponds to the union of the regions associated to each vertex of  $S$ . There exists an edge  $e = (s_i, s_j)$  between two vertices if the corresponding

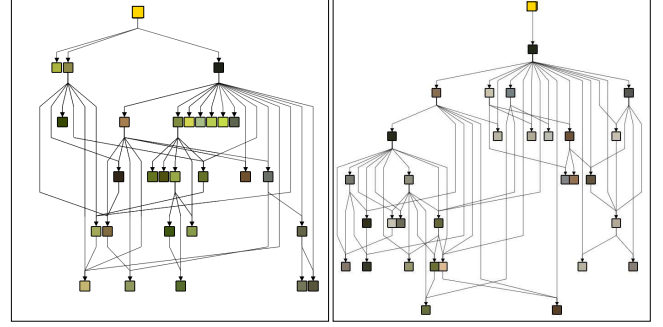


**Fig. 3.** Region adjacency graphs corresponding to segmented objects displayed in figure 2

regions  $R(s_i)$  and  $R(s_j)$  are 4-adjacent. Due to the previous remark, the RAG associated to an object may have more than one connected component. The corresponding RAGs of objects of the figure 2 are displayed<sup>1</sup> in figure 3. Here, each vertex of a RAG is represented by a squared box centered at a region’s center of gravity. The boxes are filled in with the mean color of corresponding regions in image plane. The edges depict regions’ adjacency.

The segmentation process may produce some noisy regions due to the motion mask or to the down-sampling. These regions have small area and are less relevant than regions with a high area. A consistent way for the matching should consider the biggest regions first because they represent more significant parts of the objects. Because the RAG does not capture the area of the regions, we chose to transform the RAG  $G(G_V, G_E)$  associated to a partition  $\mathcal{P}$  into a DAG denoted by  $D(D_V, D_E)$  by ordering the neighbor relations from regions with a high area to smaller regions. Since the area of the regions is close linked to the number of its neighbors, the hierarchy that we obtain with this orientation is consistent with regard to the importance of the regions in the objects. Experiments have shown that considering other orders does not improve the matching. Thus, we have  $D_V = G_V$  and  $D_E = G_E$  where the edges of  $G_E$  are directed EDGED. In a first step, each connected component of the RAG is associated with a connected component of the DAG. Let  $s$  be a vertex of  $D(D_V, D_E)$ , we denote by  $A(s)$  the relative area of its corresponding region  $R(s)$ . We define  $A(s)$  as follows:  $A(s) = \frac{|R(s)|}{|\mathcal{P}|}$ , where  $|R(s)|$  (resp.  $|\mathcal{P}|$ ) corresponds to the number of pixels of  $R(s)$  (resp.  $\mathcal{P}$ ). The inner vertices of the DAGs are the regions that have higher area than all of their neighbors. Let  $e(s, s')$  be an edge of  $G(D_V, D_E)$ , the corresponding directed edge  $e(s, s')$  in  $D(D_V, D_E)$  is oriented from  $s$  to  $s'$  iff  $A(s) > A(s')$ .

In order to have only one connected DAG for each object, we add a dummy vertex  $s_{root}$  as the root of the DAG



**Fig. 4.** Direct acyclic graphs built from the RAGs displayed in figure 3

$D(D_V, D_E)$ . We add an edge from the dummy vertex to each vertex of  $D_V$  with a null inner degree. In this way, the children vertices of  $s_{root}$  are the regions with high relative area, the leaves of this DAG are regions with the smallest areas. Note that frequently, the nearer from the root the vertex is, the higher its arity is, due to the high area of the associated regions. Now, there exists a path from the dummy vertex to all of the vertices of the DAG.

In figure 4 we show the DAGs built from the RAGs displayed in figure 3. The object associated to the left DAG is made of two connected component. Thus, the dummy vertex is connected to the vertices corresponding to the highest region of each component (two edges). The third edge links the dummy vertex to a big region that has only smaller regions as neighbors (null inner degree).

### 3. METRICS ASSOCIATED TO VERTICES

In this section, we describe several extrinsic and intrinsic metrics that will be helpful in order to predict quasi-similar parts between DAGs. We associate with each vertex  $s$  a metric vector which is based on the structural aspects. We compute the three following intrinsic metrics:

- the degree of the vertex denoted by  $\delta(s)$ ,
- the number of vertices of the sub-DAG with root  $s$  denoted by  $\mu(s)$ ,
- the so-called Strahler number of a vertex denoted by  $\sigma(s)$ .

We briefly explain this last metric. The Strahler number has first been introduced on binary trees in some works about the morphological structure of rivers [13, 18]. A generalization on planar trees has been set up [3] using a nice interpretation by Ershov [8]. He proved that the Strahler number of the root of the binary tree incremented by one is exactly the minimal number of registers needed to compute an arithmetical expression whose syntactical structure (parentheses) is encoded by the tree. Following this interpretation, for

<sup>1</sup>The RAGs are drawn with the graph visualization framework Tulip [1].

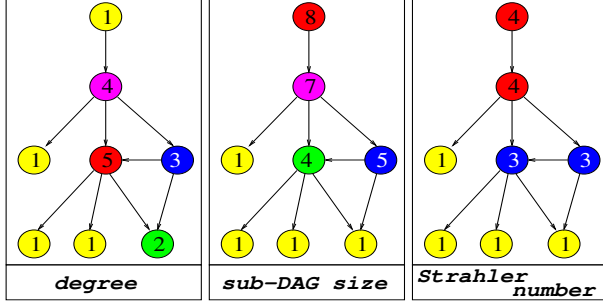


Fig. 5. Computed metrics on a DAG.

each internal vertex  $s$  having  $k + 1$  children whose roots are  $\{s_i\}_{0 \leq i \leq k}$  such that if  $i \leq j$  then  $\sigma(s_i) \geq \sigma(s_j)$ , the Strahler number  $\sigma(s)$  is given by :

$$\sigma(s) = \begin{cases} 1 & \text{if } s \text{ has no child} \\ \max_{0 \leq i \leq k} (\sigma(s_i) + i) & \text{if } s \text{ has } k + 1 \text{ children } s_i \end{cases}$$

The degree  $\delta(s)$  measures the local ramification of the vertex, and by this way if the region  $R(s)$  is adjacent to many regions, the degree will be high. The number of vertices  $\mu(s)$  captures the number of regions which are not directly adjacent to  $R(s)$  but can be reached from  $R(s)$  using a sequence of adjacent regions, with respect to the orientation of the DAG. A high Strahler number  $\sigma(s)$  means that the DAG reached from  $s$  is highly ramified. Thus in a certain sense, how the regions reachable from  $R(s)$  are parcelled out.

Note that  $\delta, \sigma$  and  $\mu$  are not in the same interval. Thus, we normalize the values in  $[0, \dots, 1]$  as following. Let  $\nu$  be a metric, the normalized value  $\tilde{\nu}(s)$  of a vertex  $s$  is given by:

$$\tilde{\nu}(s) = \frac{\nu(s) - \nu_{min}}{\nu_{max} - \nu_{min}}$$

where  $\nu_{min} = \min_{s \in D_V} \nu(s)$  and  $\nu_{max} = \max_{s \in D_V} \nu(s)$ .

Due to the structure of a DAG which is “tree-like”, these definitions are also valid on DAG.

The figure 5 shows an example of the valuation of each metrics on a DAG. A same color on vertices represents a same value.

Note that all the parameters used in this paper are invariant to usual transformations of object such as rotation, translation and scaling. Consequently, the heuristic that is based on these is robust to such transformations.

Moreover, the intrinsic parameters described above do not fully capture the complexity of the objects. Indeed, the larger the region is, the more relevant these metrics are. Since our goal is to recognize quasi-similar object extracted from images, extrinsic parameters such as the color or the surface of

the regions will be helpful information to improve the recognition based on structural data. In the section 4.2, we describe how extrinsic parameters are used to guide the recognition process.

#### 4. FINDING SIMILAR SUB-DAG

At the Infovis’03 Conference contest [9] on pairwise comparison of trees, an assigned task was to find similar sub-trees that have moved:

- the sub-trees are not in the same place in the hierarchy,
- slight changes occur between the two sub-trees

We call them *quasi-similar* sub-trees. Due to the property of DAGs (no cycle), finding “similar sub-trees in a tree” is not far away than finding “similar sub-DAGs in a DAG”. Moreover finding “similar sub-DAGs in a DAG” or “similar DAGs in several DAGs” are one and the same task. In the last case, one just needs to build a DAG with a dummy vertex (its root), which has sub-DAGs that are the DAGs to be compared. In the case of trees, works have already been done based on vertices’ degree by Zemlyachenko [21] and then by Dinitz *et al.* [7]. However, these algorithms only detect isomorphism and do not provide a measure of similarity for sub-trees. More recently, Gupta *et al.* [12] gave a nice algorithm for determining the largest tree embeddable in two trees but the complexity of their algorithm is  $O(n^2)$  (where  $n$  is the whole number of vertices of the two trees). In order to give a response to the Infovis’03 task, we have designed a heuristic [2] that can suggest, by labelling, similar parts in a tree (similar sub-trees have a same label).

Here, we adapt this heuristic in order to capture objects in the video content. In the following, we will denote by  $D(D_V, D_E)$  and  $D'(D'_V, D'_E)$  the two DAGs to be compared. The algorithm assigns labels to vertices of the two DAGs so that if vertices of two subsets  $S$  included in  $D_V$  and  $S'$  included in  $D'_V$  are identically labelled, then the associated regions  $R(S)$  and  $R(S')$  correspond to the same part of the same object.

The algorithm is in three steps :

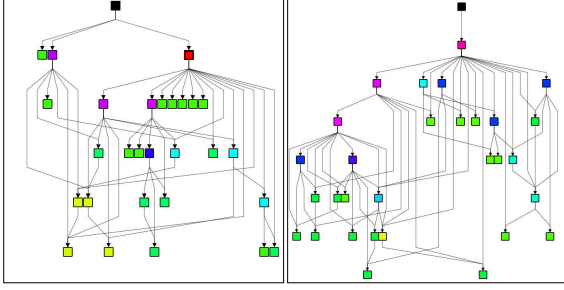
- Compute normalized intrinsic metrics for each DAG (see section 3),
- Roughly classify the vertices i.e. if two vertices in  $D$  and  $D'$  have close intrinsic metric values, label them by a same integer (section 4.1),
- Compute the final labelling  $\lambda$  by a propagation process (section 4.2).

##### 4.1. Classification of the vertices by structural similarity computation

Let  $s$  and  $s'$  be respectively in  $D_V$  and  $D'_V$  then, we label them by the same integer if

$$(\tilde{\delta}(s) - \tilde{\delta}(s'))^2 + (\tilde{\sigma}(s) - \tilde{\sigma}(s'))^2 + (\tilde{\mu}(s) - \tilde{\mu}(s'))^2 \leq \epsilon.$$





**Fig. 6.** Vertices classification by structural similarity. The label values of the vertices are mapped on a color palette. A same value corresponds to a same label.

where  $\epsilon$  is a given threshold that defines how tolerant the classification is according to the structural metrics. Note that a null value for  $\epsilon$  induces an isomorphic sub-DAGs searching. Experiments have shown that the value  $\epsilon = \frac{1}{n}$ , where  $n$  is the whole number of the vertices of  $D_V$  and  $D'_V$ , provides good results. A vertex  $v$  of  $D_V$  is not compared with all of the vertices of  $D'_V$  to find its label. In our method, we use a cover tree data structure in order to improve the computational complexity. The insertion of a new element  $v$  in this cover tree (that corresponds to the finding of its label) is in  $O(n \log(n))$ . We refer the reader to [4] for more details about this data structure.

Let  $l(s)$  be the label of a vertex  $s$ . By the classification process, we get  $l(s)$  in  $[1, \dots, l_{max}]$ . The value 1 is associated to the DAGs' sink nodes and the value  $l_{max}$  is associated to the vertices with the highest Strahler value. Note that  $l$  depends on the visit order of the vertices. Because Strahler numbers express the reachability of vertices from a vertex, we have chosen to visit the vertices in the reverse order of their Strahler numbers that is first the vertex which has the highest associated value.

Let  $S$  be a set of vertices. In the following, we will denote by  $\mathcal{F}_S(n)$  the vertices family of  $S$  labelled by a same value  $n$ . We have

$$\mathcal{F}_S(n) = \{s \in S, l(s) = n\}.$$

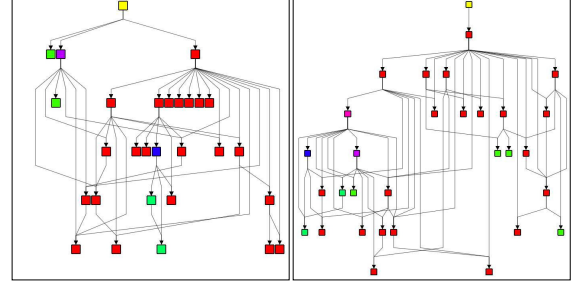
In order to simplify the notations, we will denote in the following by  $\mathcal{F}(n)$  the vertices family  $\mathcal{F}_{D_V \cup D'_V}(n)$ .

The figure 6 shows the result of the vertices classification by structural metrics similarity. A same color is used for a same label value.

#### 4.2. Matching process by propagation

In this last step, we identify patterns by incorporating children of parent vertices into the family of these parents if the children are almost similar.

After the classification step described in the previous section, if for two vertices  $s$  and  $s'$  taken from two different



**Fig. 7.** Matching of vertices by label propagation on the DAGs of the figure 6. A same color on vertices means a same label.

DAGs, the intrinsic parameters computed for  $s$  and  $s'$  are close, they have the same label  $l$ . We then infer that the associated regions  $R(s)$  and  $R(s')$  represent the same part of a same object. We propose here to compare the composition of the descent of  $s$  and  $s'$  in order to identify a quasi-similar pattern. Let  $C(s)$  (resp.  $C(s')$ ) be the set of children of  $s$  (resp.  $s'$ ). If the labels of  $C(s)$  and  $C(s')$  are almost identical, we extend the label value of the parents to their children: a quasi-similar part has been identified.

We do not only rely on the topology. We propose to consider an extrinsic parameter (the mean color of the regions) in order to reinforce the first supposition given by structural similarity of the vertices. The mean color of a region  $R(s)$  associated to a vertex  $s$  is defined in the RGB space by:

$$(\bar{R}_{R(s)}, \bar{G}_{R(s)}, \bar{B}_{R(s)})^T$$

where  $\bar{R}_{R(s)}$ ,  $\bar{G}_{R(s)}$  and  $\bar{B}_{R(s)}$  correspond to the red, the green and the blue component values of the mean color of the region  $R(s)$ .

The closer the regions are in terms of color (euclidean distance), the more tolerant the propagation process is. This means that we adjust the tolerance to the differences there exist between the labels of  $C(s)$  and  $C(s')$  by the color similarity  $\rho_{col}(s, s')$  defined as follows:

$$\rho_{col}(s, s') = 1 - \sqrt{\sum_{C \in \{R, G, B\}} (\bar{C}_{R(s)} - \bar{C}_{R(s')})^2}$$

More formally, let us build a new labelling  $\lambda$  on the vertices. At the initial step,  $\lambda$  is set to  $l$ . Let  $s$  and  $s'$  be in a same family  $\mathcal{F}(n)$ . Let  $\tau$  be a real,  $\tau \geq 1$ . Then, if, for each integer  $n'$  which labels a vertex of  $\mathcal{C}(s) \cup \mathcal{C}(s')$

$$|\text{card}(\mathcal{F}_{\mathcal{C}(s)}(n')) - \text{card}(\mathcal{F}_{\mathcal{C}(s')}(n'))| \leq \tau * \rho_{col}(s, s')$$

then for each  $v \in \mathcal{F}_{\mathcal{C}(s)} \cup \mathcal{F}_{\mathcal{C}(s')}$  we fix  $\lambda(v) = n$ .

Here, the parameter  $\tau$  fixes the structural tolerance between the children for the pattern retrieval. It defines the notion of quasi-similarity of the descent in the structural point of view.

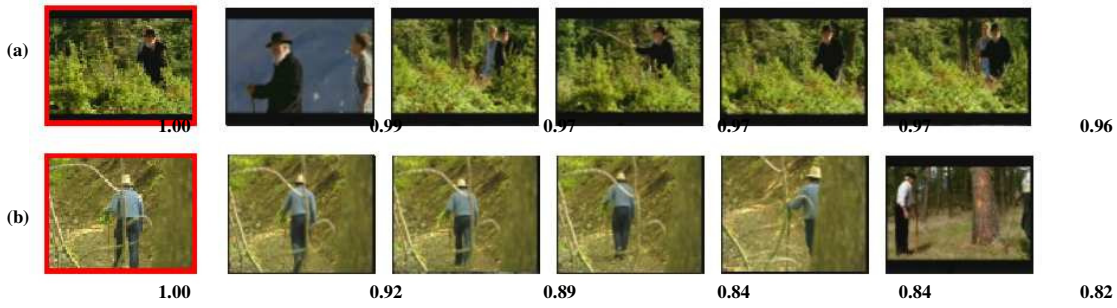


Fig. 8. Our heuristic 5 best retrievals.



Fig. 9. Method based on relaxation 5 best retrievals.

This process is done in a top-down traversal on sub-DAGs and stops as soon as  $s$  or  $s'$  is a well and all vertices have been visited. There is no backtrack that is, as soon as the label has been propagated to children, they are included in the pattern and their label will not change anymore. Of course, the visit order influences the computation. Choosing the best pair of vertices would increase drastically the complexity of the algorithm. Thus, in each DAG, the vertices are visited in a decreasing order according to the relative area  $A$  of their associated regions (see section 2 for the definition of  $A$ ).

The dummy vertices are not used in the classification process described in the previous section. Thus we label them by  $\lambda_{max} + 1$ . In this way, the propagation process begins with the two dummy vertices which represent the two objects to be compared. When all of the vertices of a family of label  $n$  have been visited (and recursively the children in the case of matching), the process continues by considering unmarked vertices of the next family (label  $n - 1$ ) until all vertices have been visited for matching.

Note that the retrieval is not based on the matching of the dummy vertices of the DAGs. The process aims to recognize patterns (sub-DAGs) into the DAGs. When two similar parents propagate their label to their children, both parents and children are marked as matched vertices.

The figure 7 illustrates the result of the propagation process applied on the DAGs displayed on figure 6. Colors represent the different families of nodes (the color of the parents

has been propagated to the children). The red colored parts of the DAGs corresponds to the quasi-similar pattern that has been identified between the two sample DAGs of figure 6.

## 5. SIMILARITY MEASURE OF OBJECTS

The similarity measure we use in this paper corresponds to a size evaluation of the part of objects that have been identified as quasi-similar. Let  $D$  and  $D'$  be two DAGs that represent objects to be compared. Let  $S$  and  $S'$  be the vertices of  $D$  and  $D'$  respectively corresponding to the marked vertices (vertices identified as belonging to similar pattern). Remind that a vertex is marked when, during the label propagation process, it is considered in a label propagation (as a parent if it propagates his label to children or as a child if it takes the label of its parent).

The similarity measure  $\theta(D, D')$  between the objects represented by the DAGs  $D$  and  $D'$  is defined as follows:

$$\theta(D, D') = \frac{1}{2} \left( \sum_{s \in S} A(s) + \sum_{s' \in S'} A(s') \right)$$

We recall that  $A(s)$  (introduced in section 2) corresponds to the relative area of the region  $R(s)$  associated to the vertex  $s$  according to the whole area of the object partition.

The similarity measure  $\theta$  evaluates the area of objects that has been matched. This means that we first compute the whole relative area of matched regions for each set  $S$  and  $S'$ .

The object similarity measure then corresponds to the mean of these two values.

This measure is used to order the objects contained in the video database by similarity with a query object request.

## 6. RESULTS

We have tested our method for objects retrieval in sequences at DC-resolution taken from CERIMES © MPEG2 compressed documentaries. The segmented objects are extracted from DC-frames of size  $76 \times 92$  pixels and at the temporal resolution of two frames per second.

The sequences are taken from CERIMES © documentary videos *Aquaculture en méditerranée*, *De l'arbre à l'ouvrage*, *Le chancre* and *Hiragasy* and contain about 5,000 frames from which objects have been extracted. For the experiments, 100 objects corresponding to people have systematically been chosen randomly from the video objects database.

We have evaluated the performance of our method in the context of query by example. Retrieval systems often present query by example results in terms of  $k$  best matches [10, 20]. A match is correct if the object represents the query. Two examples of object retrieval are shown in figure 8. The scores under frames correspond to the object similarity measure  $\theta$  as defined in section 5. The example (a) illustrates the ability of our method to retrieve the same object under different conditions: the similarity measures are good even if the same old man appears in two different shots. Note that the scores obtained for the objects taken from the same shot than the query are less than the best match which is extracted from another scene. This is due to the quality of the motion mask that defines the region of interest. The zone of interest is automatically computed by a motion analysis [16] and does not exactly correspond to the foreground object contained in the frame (static parts of objects may be not detected by the motion detection and small background regions that was occluded in the previous frames are often included into the motion mask). In this way, the recognition method will not be able to correctly recognize objects because of the inaccuracy of the motion mask. In the example (b), the four best responses are relevant. The fifth does not represent the same object. However, the structures of the two considered objects (standing men with dark trousers and bright skirt) are very close to each other. The topology of the objects are similar enough not to be disturbed by the color tolerance coefficient used in this article.

The interest of considering local neighborhoods for region matching process has been shown in our previous work that uses relaxation techniques [5]. In the paper, starting from an initial similarity measure between pairs of vertices, we iteratively update by increasing or decreasing the similarity value according to the likeness of their neighborhoods.

In [5], the strategy consists in the use of the local structure of the objects to refine a similarity measure based on regions

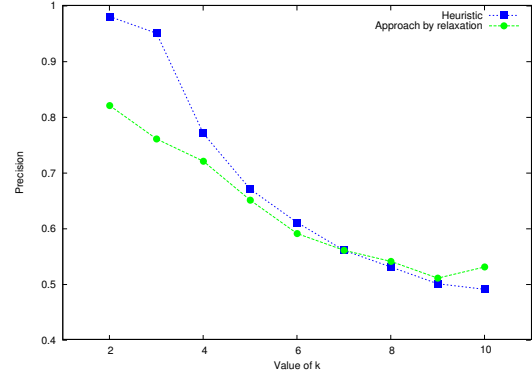


Fig. 10. Object retrieval precision for different values of  $k$ .

features. The heuristic defined here proposes to reverse the problem. It begins by capturing a structure similarity and it drives the propagation process using the regions visual features.

We have compared the method based on relaxation techniques [5] with the approach proposed in this paper. The precision figures for different values of the number of best matches  $k$  for both methods are plotted in figure 10. Precision is computed as being the ratio between the number of correct matches and  $k$ .

Both approaches provide comparable results. The heuristic is more precise for the three first responses whereas the relaxation offers a better precision for more than 8 responses. In [5], the whole topology of objects is not taken into account and two large regions that are close enough to be matched, can imply a high object similarity. These problems are avoided in the heuristic approach because both of global topology of the object, local neighborhood and color features of the regions are used to identify common patterns between the two objects we compare.

The heuristic is not altered by usual deformations such as rotation, translation and scaling because the structure of objects is invariant to these. It is also robust to image alteration (contrast and luminosity variation, blur, noise) because the only color similarity parameter is altered by these changes.

## 7. CONCLUSION

In this paper, we have presented a new approach to the problem of object matching recognition in video in the context of the rough indexing paradigm. In this context, classical methods mainly based on region features are inefficient because image data are scarce due to the down-sampling. This lack of information requires to consider the structure of the object as the most relevant information. Therefore, we use intrinsic parameters in order to compare the structure of the DAGs associated with segmented objects. The vertices with a same label in the classification process have a quasi-similar structure. The prolongation of the labelling function is driven by color



similarity between regions associated to vertices. In this way, the visual similarity between regions allows us to be more tolerant to structural differences.

This approach offers good results in the rough indexing paradigm. The domain of application of this methods may be: retrieval of video shots that contain a given object, semantic inventory of video shots into video chapters or scenes.

Now, we plan to investigate our method for image in full resolution. The scheme of algorithm will stay the same for the structural labelling, concerning the prolongation we have to define the visual feature vector that will be more complete than the one used for rough data. Moreover, we have to tune the threshold  $\tau$  to adapt the heuristic to a such resolution.

## 8. REFERENCES

- [1] D. Auber. *Graph drawing software*, chapter Tulip - A Huge Graph Visualization Framework. Verlag, 2003.
- [2] D. Auber, M. Delest, J.P. Domenger, P. Ferraro, and R. Strandh. EVAT : Environment for visualization and analysis of trees. In *IEEE Symposium on Information Visualisation Contest*, volume [www.cs.umd.edu/hcil/iv03contest/](http://www.cs.umd.edu/hcil/iv03contest/), pages 124–126, 2003.
- [3] D. Auber, M. Delest, J.M. Fédou, J.P. Domenger, and P. Duchon. New strahler numbers for rooted plane trees. In M. Drmota, P. Flajolet, D. Gardy, and B. Gittenberger, editors, *Third Colloquium on Mathematics and Computer Science, Algorithms, Trees, Combinatorics and Probabilities*, Trends in Mathematics, pages 203–215. Vienna University of Technology, Birkhauser, 2004.
- [4] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *ACM International Conference Proceeding Series, Proceedings of the 23rd international conference on Machine learning*, volume 148, pages 97–104, Pittsburgh, Pennsylvania, 2006. ACM Press.
- [5] F. Chevalier, J.P. Domenger, J. Benois-Pineau, and M. Delest. Retrieval of objects in video by similarity based on graph matching. *Pattern Recognition Letters*, 2007. To appear.
- [6] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.
- [7] Y. Dinitz, A. Itai, and M. Rodeh. On an algorithm of Zemlyachenko for subtree isomorphism. *Information Processing Letters*, 70(3):141–146, 1999.
- [8] A.P. Ershov. On programming of arithmetic operations. *Communication of the A.C.M.*, 1(8):3–6, 1958.
- [9] J.D. Fekete and C. Plaisant. Infovis contest 2003 - visualization and pair wise comparison of trees. In IEEE, editor, *IEEE Symposium on Information Visualization*, volume [www.cs.umd.edu/hcil/iv03contest/](http://www.cs.umd.edu/hcil/iv03contest/), 2003.
- [10] M. Flickner, H. Sawhney, W. Niblack, and al. Query by image and video content: The qbic system. *IEEE Computer*, 28:23–32, September 1995.
- [11] C. Gomila and F. Meyer. Graph-based object tracking. In *International Conference on Image Processing*, September 14-17 2003.
- [12] A. Gupta and N. Nishimura. Finding largest subtrees and smallest supertrees. *Algorithmica*, 21(2):183–210, 1998.
- [13] R.E. Horton. Eroded development of systems and their drainage basins, hydrophysical approach to quantitative morphomology. *Bulletin Geological Society of America*, 56:275–370, 1945.
- [14] B. Huet and E.R. Hancock. Inexact graph retrieval. In *IEEE CVPR99 Workshop on Content-based Access of Image and Video Libraries (CBAIVL-99)*, pages 40–44, Fort Collins, Colorado USA, June 22 1999.
- [15] J. Kittler, W. J. Christmas, and M. Petrou. Probabilistic relaxation for matching problems in computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(5):617–623, September 1985.
- [16] F. Manerba, J. Benois-Pineau, and R. Leonardi. Real-time rough extraction of foreground objects in mpeg1,2 compressed video. In *Proc. Workshop on Image Analysis For Multimedia Interactive Services (WIAMIS)*, Montreux, Switzerland, April 2005.
- [17] W. Seales, C. Yuan, W. Hu, and M. Cutts. Object recognition in compressed imagery. *Image and Vision Computing*, 16(5):337–352, April 1998.
- [18] A.N. Strahler. Hypsomic analysis of erosional topography. *Bulletin Geological Society of America*, 63:1117–1142, 1952.
- [19] J. R. Ullman, V. Sridhar, and X. Li. An algorithm for subgraph isomorphism. *Journal of the ACM*, 23(1):31–42, 1976.
- [20] J.Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.
- [21] V.N. Zemlyachenko. Determining tree isomorphism. *Seminar on Combinatorial Mathematics*, pages 54–60, 1971.